Package: digiNORM (via r-universe)

May 16, 2026

Type Package

Title Data-Driven Digital PCR Normalization

Version 0.1.0

Description Adopts the general least squares-based data-driven normalization strategy developed by Heckmann et al. (2011) <doi:10.1186/1471-2105-12-250> to correct for technical variance in gene expression data generated via digital polymerase chain reaction (dPCR). Performs normalization of raw copy numbers and also calculates relative variability metrics that can be used to assess the impact of normalization on variance.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports utils

Suggests testthat (>= 3.0.0)

Author Grant C. O'Connell [aut, cre] (ORCID: <https://orcid.org/0000-0002-2642-6670>)

Maintainer Grant C. O'Connell <goconnell.phd@gmail.com>

Depends R (>= 3.5)

Config/testthat/edition 3

NeedsCompilation no

Repository https://cran.r-universe.dev

Date/Publication 2026-04-16 21:07:01 UTC

RemoteUrl https://github.com/cran/digiNORM

RemoteRef HEAD

RemoteSha 1a6e3ef34d52ffd28a19c15e5a4680890486e56f

Contents

`digiNORM-package` *Data-Driven Digital PCR Normalization*

Description

The 'digiNORM' package enables normalization of raw gene expression data generated via digital polymerase chain reaction (dPCR). Normalization is carried out using an application-specific adoption of the least squares-based data-driven strategy developed by Heckmann et al. (2011), and previously applied for traditional quantitative reverse transcription polymerase chain reaction (qRT-PCR) data in the 'NORMAgene' package written by O'Connell (2026). The 'digiNORM' package employes an identical core normalization algorithm as the 'NORMAgene' package; it uses within experimental condition least squares fits to estimate per-replicate technical variance and generate corresponding multiplicative correction factors that are ultimately applied for normalization. Normalization does not rely on expression information from reference transcripts, and can be carried out on data from as few as five target transcripts of interest. However, relative to the 'NORMAgene' package, additional automated processing is internally implemented both upstream and during normalization to accommodate features unique to count-based dPCR data, including steps to facilitate handling of zero counts.

Details

The primary user-facing function is `digi_norm()`, which is suitable for most standalone single experiment normalization workflows. `digi_norm()` applies the core normalization algorithm to raw dPCR copy numbers provided via an input data frame appended with requisite experimental metadata, and outputs an identically structured data frame containing normalized values. Given that normalization is based on least squares fits, stable normalization requires information from a minimum five of target transcripts with non-zero values in a majority of replicates within each experimental condition. While additional data from more sparsely detected targets may be present and inform normalization, under default settings, automated within-condition weighting based on detection rate is implemented to prioritize information from targets with a higher proportion of non-zero data when calculating correction factors. It is important to note that in situations where data from targets with zero-inflated copy numbers are present, even with detection rate-based weighting, normalization is more likely to be reliable when the overall target-wise patterns of detection are relatively consistent between replicates within each experimental condition.

Given that robust normalization is dependent on access to appropriate information, in addition to generating normalized copy numbers, digi_norm() also calculates two diagnostic metrics that allow users to evaluate the suitability of the input data according to the general guidelines outlined in the prior paragraph. The first metric is the number of informative targets, which represents the number of target transcripts in the input data for which non-zero copy numbers are present in at least 75% of replicates. The number of informative targets is calculated within experimental conditions, and summarized cumulatively across all experimental conditions, with the later value representing the total number of target transcripts registered as informative at least once. The second is detection concordance, which represents the average Jaccard similarity calculated between all pairwise combinations of replicates with respect to the presence or absence of non-zero copy numbers across targets. Values range from 0 to 1, with larger values indicating a higher degree of homogeneity in target-wise detection patterns between replicates. Detection concordance is calculated within experimental conditions and summarized cumulatively across all experimental conditions via simple average.

Beyond the aforementioned metrics focused on assessing the properties of the input data, digi_norm() also calculates an additional diagnostic metric, relative variability, which users can employ to directly evaluate the ultimate effect of normalization on copy number variance. This metric is identical the relative variability metric calculated by the 'NORMAgene' package, and represents the proportional change in log-space copy number standard deviation pre to post normalization. Values of less than 1 indicate a reduction in variance as a result of normalization, and values of greater than 1 indicate an increase in variance as a result of normalization. Relative variability values are calculated at the level of individual target transcripts within experimental conditions, and are further summarized cumulatively at the condition and cross-condition levels by simple averages.

For a given normalization, summary.digi_norm() can be used to print a summary which includes the number of informative targets, detection concordance, and a list of targets that were more the most heavily weighted in the correction factor calculation, as well as high-level relative variability information. The exact correction factors applied for normalization can be accessed using correction_factors(), while more detailed weighting and relative variability metrics can be accessed using normalization_weights() and relative_variability().

Note: digi_norm_core() provides matrix-based execution of the core normalization algorithm and is internally called by digi_norm(). While use of digi_norm() is recommended in a majority of situations, directly calling digi_norm_core() may afford advanced users a lightweight option for cleaner integration into larger post-analytical pipelines. digi_norm_core() is not exported and can only be called via the internal namespace operator.

Two real-world dPCR datasets generated by the O'Connell laboratory at Case Western Reserve University (Cleveland, OH, USA) are also included, which are used in the documentation examples. The dataset multi_cond_data contains raw copy numbers and experimental meta-data from an intra-animal comparison of gene expression between five anatomically distinct murine brain regions. It can be used to demonstrate or evaluate normalization workflows for use-cases involving data from multiple experimental conditions. The dataset single_cond_data contains raw copy numbers and experimental meta-data from a single cohort study of murine skeletal muscle gene expression. It can be used to demonstrate or evaluate normalization workflows for use-cases involving data from a single experimental condition.

Main functions

digi_norm() Normalize raw copy numbers stored in a data frame.

summary.digi_norm() Summarize digiNORM normalization.

correction_factors() Retreive per-replicate correction factors.

normalization_weights() Retreive target weights applied in correction factor calculation.

relative_variability() Retreive relative variability metrics.

Datasets

multi_cond_data Example dataset from a real-world multi-condition experiment.

single_cond_data Example dataset from a real-world single condition experiment.

Citation

If you use the 'digiNORM' package in published work, please cite:

O'Connell, GC. (2026). *digiNORM*. R package version 0.1.0. Available from https://CRAN.R-project.org/package=digiNORM.

References

Heckmann, LH., Sørensen, PB., Krogh, PH., & Sørensen, JG. (2011). NORMA-Gene: a simple and robust method for qPCR normalization based on target gene data. *BMC Bioinformatics*, 12, 250. doi:10.1186/1471210512250

O'Connell, GC. (2026). *NORMAgene*. R package version 0.1.1. Available from https://CRAN.R-project.org/package=NORMAgene.

See Also

digi_norm()
summary.digi_norm()
correction_factors()
normalization_weights()
relative_variability()
multi_cond_data
single_cond_data

---

correction_factors *Retrieve correction factors from digiNORM output*

Description

Retrieves the per-replicate multiplicative correction factors used for normalization.

Usage

```
correction_factors(object)
```

Arguments

object	An object returned by digi_norm().

Value

A numeric vector of correction factors. If replicate identifiers were passed to digi_norm(), the vector is named accordingly.

See Also

digi_norm()
digiNORM-package

Examples

```
# load example dataset containing raw copy numbers
# and metadata from a multi-condition experiment

data(multi_cond_data)
raw_data <- multi_cond_data

#normalize copy numbers

norm_data <- digi_norm(
  data = raw_data,
  conditions = "Brain_region",
  replicates= "Sample_id"
)

# retrieve correction factors

correction_factors(norm_data)
```

---

digi_norm *Normalize copy numbers using digiNORM*

Description

Applies least squares-based data-driven normalization to raw dPCR copy numbers provided via an input data frame appended with experimental meta-data. Returns a data frame containing normalized copy numbers with informative target metrics, detection concordance metrics, target weights, correction factors, and relative variability metrics attached as attributes. Raw copy numbers can be provided in the form of either positive partition counts or single molecule counts calculated using the Poisson distribution.

Usage

```
digi_norm(
  data,
  conditions = NULL,
  replicates = NULL,
  targets = NULL,
  weight_by_detection = TRUE,
  weight_factor = 2,
  weight_zero = NULL,
  weight_resolution = 100,
  show_warnings = TRUE
)
```

Arguments

data	A data frame structured with biological replicates in rows, and experimental metadata and target-wise raw copy numbers in columns.
conditions	A single column name in data specifying experimental condition membership in the case of a multi-condition experiment, or NULL in the case of a single condition experiment. Normalization is applied within experimental conditions when specified, or across all replicates when NULL. This argument must be explicitly provided.
replicates	A single column name in data containing replicate identifiers, or NULL if replicate identifiers are not present. If provided, replicate identifiers are used for naming of outputs only, and are not used in normalization calculations. This argument must be explicitly provided.
targets	Optional character vector specifying target transcripts to be normalized. All items must be column names in data containing raw copy numbers. If NULL, all numeric columns except conditions and replicates are used.
weight_by_detection	Specifies whether to weight target transcripts based on detection rate when calculating correction factors. If FALSE, all targets automatically detected in data or specified in targets equally contribute to correction factor calculation unless otherwise indicated by weight_zero. If TRUE, all targets are assigned weights between 0 and 1 based on the proportion of replicates with non-zero copy numbers present within each experimental condition raised to the power of weight_factor, which are subsequently used when calculating correction factors. Targets with calculated weights of less than 0.01 within a given condition are assigned a final weight of 0 and dropped from correction factor calculation for said condition. Default value is TRUE.
weight_factor	Numeric value ranging from 1 to 10 specifying the penalty to apply for non-detection when calculating target transcript weights when weight_by_detection is TRUE, with larger values producing larger downweighing for non-detection. Default value is 2.
weight_zero	Optional character vector specifying target transcripts to exclude from correction factor calculation. All items must be column names in data automatically

detected or specified by `targets` to contain target transcript copy numbers. Targets specified in `weight_zero` will still be normalized using the final correction factors. If `NULL`, all targets automatically detected in `data` or specified by `targets` will be used for correction factor calculation unless empirically dropped due to detection rate-based weighting.

`weight_resolution`
: Numeric value ranging from 10 to 1000 controlling at how fine a resolution the calculated target weights are applied when calculating correction factors when `weight_by_detection` is `TRUE`. Weighting is applied via target-copy number duplication, so larger values will result in higher more precise application of weights but higher computational burden. Default value is 100.

`show_warnings`
: Specifies whether to print warnings generated during normalization. Default value is `TRUE`.

Details

Users must explicitly specify how experimental conditions and replicate identifiers are handled to avoid accidental normalization of numeric metadata. Because the multiplicative correction factors applied for normalization are calculated within experimental conditions, accurate experimental meta-data is needed for valid normalization. Correction factors can be retrieved from the output object using `correction_factors()`. Final target weights can be retrieved from the output object using `normalization_weights()`. Full relative variability metrics can be retrieved from the output object using `relative_variability()`. The number of informative targets and detection concordance, along with a summary of high-level target weight and relative variability information, can be printed using `summary.digi_norm()`. For more information on the normalization algorithm itself, or interpreting informative target, detection concordance, or relative variability metrics, see digiNORM-package.

Value

A data frame with the same organization as `data` containing normalized copy numbers, and any provided experimental metadata. The per-replicate correction factors used for normalization are attached as an attribute, as are the final target weights used for correction factor calculation, informative target metrics, detection concordance, and relative variability metrics.

See Also

`summary.digi_norm()`
`correction_factors()`
`normalization_weights()`
`relative_variability()`
digiNORM-package

Examples

```
# USE-CASE WITH MULTIPLE EXPERIMENTAL CONDITIONS

# load example dataset containing raw copy numbers
# and metadata from a multi-condition experiment
```

```
data(multi_cond_data)
raw_data <- multi_cond_data

#normalize copy numbers using digiNORM

norm_data<-digi_norm(
  data = raw_data,
  conditions = "Brain_region",
  replicates= "Sample_id"
)

# summarize normalization

summary(norm_data)

# USE-CASE WITH a SINGLE EXPERIMENTAL CONDITION

# load example dataset containing raw copy numbers
# and metadata from a single-condition experiment

data(single_cond_data)
raw_data<-single_cond_data

#normalize copy numbers using digiNORM

norm_data<-digi_norm(
  data = raw_data,
  conditions = NULL,
  replicates= "Sample_id"
)

# summarize normalization

summary(norm_data)
```

`multi_cond_data` *Example dataset from a multi-condition dPCR experiment.*

Description

A real-world dPCR generated by the O'Connell laboratory at Case Western Reserve University (Cleveland, OH, USA). The dataset contains raw copy numbers for 10 transcripts measured in total RNA isolated from intra-donor matched biopsies harvested from 8 anatomically distinct brain regions of 8 adult C57BL/6 mice. Copy numbers are in the form of transcripts per nanogram (ng) of input as calculated by the Poisson distribution, and were measured via the QIAquity One platform (Qiagen GmbH, Hilden, Germany). NA values are missing at random as a result of failed partitioning quality control.

Format

A data frame structured with biological replicates in rows, replicate identifiers in a single column, brain region in a single column, and raw copy numbers for each of the 10 target transcripts in the remaining columns.

Details

This dataset is suitable for demonstrating or evaluating normalization workflows for use-cases involving data from multiple experimental conditions.

Examples

```
#load example dataset

data(multi_cond_data)

#return dataset structure

str(multi_cond_data)
```

---

normalization_weights *Retrieve target weights from digiNORM output*

Description

Retrieves the weights assigned to each target transcript when calculating the correction factors used for normalization.

Usage

```
normalization_weights(object)
```

Arguments

object	An object returned by digi_norm().

Value

A named numeric matrix containing the final target weights used for calculation of correction factors within each experimental condition, and summarized cumulatively across all experimental conditions via simple averages.

See Also

digi_norm()
digiNORM-package

Examples

```
# load example dataset containing raw copy numbers
# and metadata from a multi-condition experiment

data(multi_cond_data)
raw_data <- multi_cond_data

#normalize copy numbers

norm_data <- digi_norm(
  data = raw_data,
  conditions = "Brain_region",
  replicates= "Sample_id"
)

# retrieve target weights

normalization_weights(norm_data)
```

`relative_variability` *Retrieve relative variability metrics from digiNORM output*

Description

Retrieves relative variability metrics calculated during normalization.

Usage

```
relative_variability(object, type = c("by_target", "by_condition"))
```

Arguments

`object`	An object returned by `digi_norm()`.
`type`	Character string specifying which relative variability metric to return. One of `"by_target"` or `"by_condition"`.

Details

For more information on interpreting relative variability metrics, see digiNORM-package.

Value

Depending on `type`:

by_target A named numeric matrix of relative variability values, calculated for each target transcript within experimental conditions, and summarized cumulatively for each target transcript across all experimental conditions by simple averages.

by_condition A named numeric vector of relative variability values summarized across all target transcripts at the condition level, as well as cumulatively across all condition levels, both via simple averages.

See Also

digi_norm()
digiNORM-package

Examples

```
# load example dataset containing raw copy numbers
# and metadata from a multi-condition experiment

data(multi_cond_data)
raw_data <- multi_cond_data

#normalize copy numbers

norm_data <- digi_norm(
  data = raw_data,
  conditions = "Brain_region",
  replicates= "Sample_id"
)

# retrieve relative variability metrics

relative_variability(norm_data, type = "by_target")
relative_variability(norm_data, type = "by_condition")
```

---

single_cond_data *Example dataset from a single condition dPCR experiment.*

Description

A real-world dPCR generated by the O'Connell laboratory at Case Western Reserve University (Cleveland, OH, USA). The dataset contains raw copy numbers for 15 transcripts measured in total RNA isolated from skeletal muscle biopsies harvested from a single cohort of 10 adult C57BL/6 mice. Copy numbers are in the form of transcripts per nanogram (ng) of input as calculated by the Poisson distribution, and were measured via the QIAquity One platform (Qiagen GmbH, Hilden, Germany). NA values are missing at random as a result of failed partitioning quality control.

Format

A data frame structured with biological replicates in rows, replicate identifiers in a single column, and raw copy numbers for each of the 15 target transcripts in the remaining columns.

Details

This dataset is suitable for demonstrating or evaluating normalization workflows for use-cases involving data from a single experimental condition.

Examples

```
#load example dataset

data(single_cond_data)

#return dataset structure

str(single_cond_data)
```

`summary.digi_norm` *Summarize digiNORM normalization*

Description

Provides a concise human readable summary of normalization performed by `digi_norm()`, including the number of informative targets and detection concordance, along with high-level target weight and relative variability information.

Usage

```
## S3 method for class 'digi_norm'
summary(object, ...)
```

Arguments

`object`	An object returned by `digi_norm()`.
`...`	Further arguments passed to or from other methods.

Details

No values are recomputed; all values are extracted from the stored normalization results. For more information on the normalization algorithm itself, or interpreting informative target, detection concordance, or relative variability metrics, see digiNORM-package.

Value

A console printed summary including:

The total number of replicates (samples), target transcripts, and experimental conditions parsed from the input during normalization.

The total number of replicates associated with each individual experimental condition.

The number of informative targets calculated for each experimental condition, and summarized cumulatively across all experimental conditions, with the cumulative value representing the number of target transcripts deemed as informative in at least one condition.

The detection concordance calculated for each experimental condition, and summarized cumulatively across all experimental conditions by simple average.

Weights associated with the top 10 most heavily weighted target transcripts used for calculation of correction factors, summarized cumulatively across all experimental conditions via simple averages.

Relative variability values summarized across all target transcripts at the condition level, as well as cumulatively across all condition levels, both by simple averages.

Warning flags associated with informative target and detection concordance metrics that could result in unstable normalization if applicable.

See Also

digi_norm()
digiNORM-package

Examples

```
# load example dataset containing raw copy numbers
# and metadata from a multi-condition experiment

data(multi_cond_data)
raw_data <- multi_cond_data

#normalize copy numbers

norm_data <- digi_norm(
  data = raw_data,
  conditions = "Brain_region",
  replicates= "Sample_id"
)

# summarize normalization

summary(norm_data)
```

Index